Генетический алгоритм: описание

Генетический алгоритм работает с представленными в конечном алфавите строками **S** конечной длины **l**, которые используются для кодировки исходного множества альтернатив **W**. Строки представляют собой упорядоченные наборы из **l** элементов: $S=(s_1, s_2, ..., s_l)$, каждый из которых может быть задан в своём собственном алфавите V_i, $i=\overline{1,L}$, т.е. s_{iO} V_i, $i=\overline{1,L}$, где алфавит V_i является множеством из r_i символов: $V_i=\{v_{ij}, j=\overline{1,r_i}\}$. Для решения конкретной задачи требуется однозначно отобразить конечное множество альтернатив **W** на множество строк подходящей длины **(**очевидно, что длина строк зависит от алфавитов, используемых для их задания**).**

Для работы алгоритма необходимо на множестве строк $U^m_{(V_1,V_2,...,V_m)}$ задать неотрицательную функцию **F(S),** определяющую показатель качества, "ценность" строки **S**O $U^m_{(V_1,V_2,...,V_m)}$. Алгоритм производит поиск строки, для которой $F^*(S)=\arg\max_{S\in U^m_{(V_1,V_2,...,V_m)}} F(S)$

Если на множестве **W** задана целевая функция **f(w),** то функцию **F(S)** на множестве строк $U^m_{(V_1,V_2,...,V_m)}$ можем определить следующим образом: **F(S)=f(w),** если элемент **w** при отображении исходного множества **W** на множество строк был сопоставлен строке **S.**

Генетический алгоритм за один шаг производит обработку некоторой популяции строк. Популяция **G(t)** на шаге **t** представляет собой конечный набор строк: $G(t)=(S_1^t, S_2^t, ..., S_N^t)$, $S_k^t\in U^m_{(V_1,V_2,...,V_m)}$, $k=\overline{1,N}$, где **N --** размер популяции, причём строки в популяции могут повторяться.

Анализ работы алгоритма удобно производить, используя аппарат схем. Схемой в генетическом алгоритме называют описание некоторого подмножества строк. Схема $H=(h_1, h_2, ..., h_m)$ может рассматриваться как строка, алфавиты для элементов которой дополнены специальным символом "#": $H\in U^m_{(V_1^H,V_2^H,...,V_m^H)}$, $V_i^H=V_i\cup"\#"$. Если в некоторой позиции **r** схемы **H** присутствует символ "#", то такая позиция называется свободной, а сам символ "#" интерпретируется как произвольный символ из алфавита V_r. Позиция **q** схемы **H** называется фиксированной, если в этой позиции присутствует один из символов алфавита V_q. Схема **H,** в которой определены фиксированные и свободные позиции, описывает подмножество $U_H\subseteq U^m_{(V_1,V_2,...,V_m)}$, содержащее такие строки, у которых элементы, соответствующие фиксированным позициям схемы, совпадают с символами схемы, а элементы, соответствующие свободным позициям схемы, являются произвольно заданными в соответствующих алфавитах: $U_H=\left\{S \mid S\in U^m_{(V_1,V_2,...,V_m)}\wedge(\forall i(i\in I_{[1,m]}\wedge h_i\neq"\#")\rightarrow(s_i=h_i))\right\}$ где $I_{[1,\ m]}$ - множество целых чисел отрезка **[1, m].**

Например, для множества строк $U^5_{(V_1,V_2,V_3,V_4,V_5)}$, где $V_i=\{0, 1\}$, $k=\overline{1,5}$, схема H_1 **="1###0"** задаёт такое множество строк, у которых первым элементом является символ **"1",** пятым **- "0",** а остальные **-** либо **"0",** либо **"1".** Строки **"10010", "11110"** являются примерами строк, принадлежащих множеству U_{H_1}.

Часть популяции $G(t)=(S_1^t, S_2^t, ..., S_N^t)$, строки которой удовлетворяют схеме **H,** обозначают $G_H(t)=(S_1^{H,t}, S_2^{H,t}, ..., S_{n(H,t)}^{H,t})$, где **n(H, t) -** число строк схемы **H** в популяции **G(t)** и называют подпопуляцией, соответствующей схеме **H.**

Суть генетического алгоритма заключается в следующем.

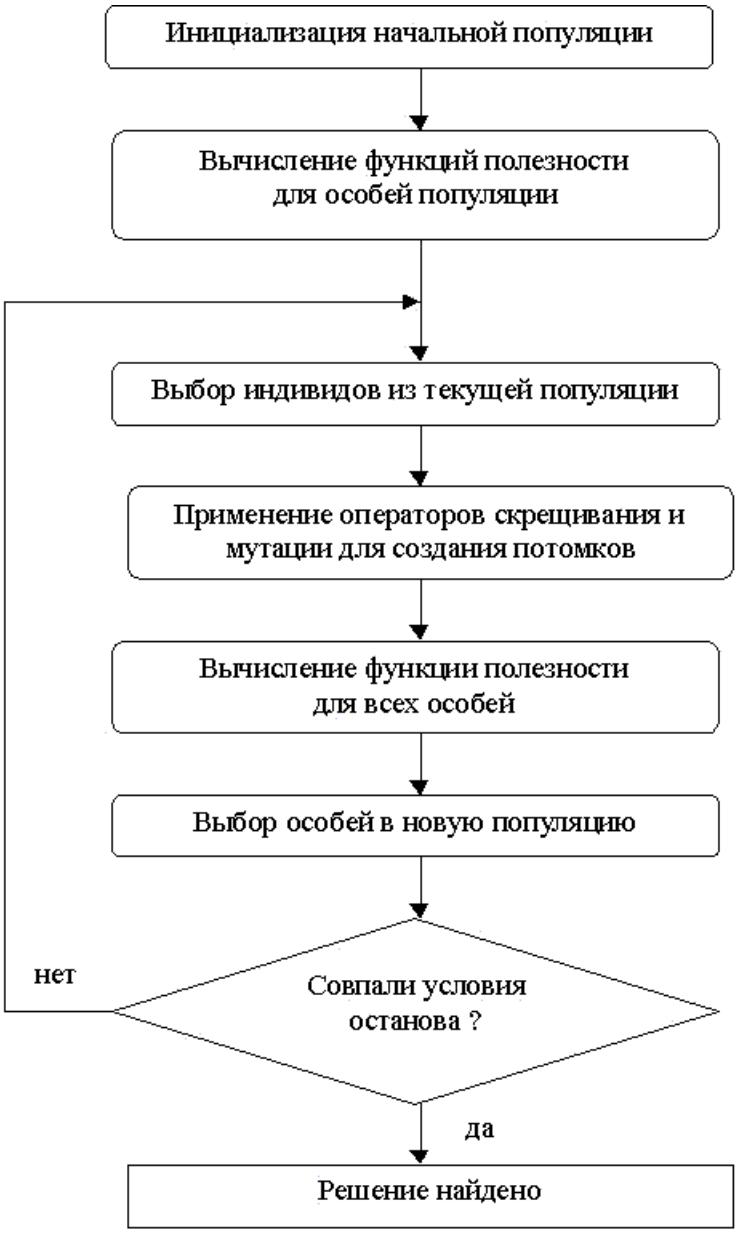
Пусть на шаге **t** имеется популяция **G(t),** состоящая из **N** строк. Для популяции вводится понятие средней ценности популяции F_{cp} **(G(t)):** $F_{cp}(G(t))=\frac{1}{N}\sum_{k=1}^{N}F(S_k^t)$

Аналогично для подпопуляции $G_H(t)$, удовлетворяющей схеме **H,** вводится понятие средней ценности подпопуляции F_{cp} $(G_H(t))$: $F_{cp}(G_H(t))=\frac{1}{n(H,t)}\sum_{k=1}^{n(H,t)}F(S_k^{H,t})$.

Генетический алгоритм осуществляет переход от популяции **G(t)** к популяции **G(t+1)** таким образом, чтобы средняя ценность составляющих её строк увеличивалась, причём количество новых строк в популяции равно **K**Ч **N**, где **K** - коэффициент новизны. Если **K<1**, то популяция будет перекрывающейся, т.е. в новой популяции сохраняются некоторые строки из старой, а если **K=1**, то она будет неперекрывающейся, т.е. подвергнется полному обновлению.

Генетический алгоритм включает три операции: воспроизводство, скрещивание, мутация.

Блок-схема генетического алгоритма



Генетический алгоритм: основные операции

Генетический алгоритм включает три операции: воспроизводство, скрещивание, мутация.

Воспроизводство представляет собой процесс выбора **K*N N** строк популяции **G(t)** для дальнейших генетических операций. Выбор производится случайным образом, причём вероятность выбора строки S_i^t пропорциональна её ценности:

$$p_{выб}(S_i^t) = \frac{F(S_i^t)}{\sum_{k=1}^{N} F(S_k^t)}$$

Процесс выбора повторяется **K*N N** раз. Предполагаемое количество экземпляров строки S_i^t в популяции **G(t+1)** равно $n_{выб}(S_i^t) = p_{выб}(S_i^t) \cdot K \cdot N$

Операция воспроизводства увеличивает общую ценность последующей популяции путём увеличения числа наиболее ценных строк.

Пусть в популяции **G(t)** содержится **n(H, t)** строк, удовлетворяющих схеме **H**. Тогда в результате воспроизводства количество строк, удовлетворяющих схеме **H** в популяции **G(t+1)** будет равно **n(H, t+1)**:

$$n(H,t+1) = \sum_{i=1}^{n(H,t)} \left(K \cdot N \cdot p_{выб}(S_i^{H,t})\right) = K \cdot N \cdot \frac{\sum_{i=1}^{n(H,t)} F(S_i^{H,t})}{\sum_{j=1}^{N} F(S_j^t)} \quad \textbf{. (1)}$$

Используя выражения для средней ценности популяции $F_{ср}(G(t))$ и подпопуляции $F_{ср}(G_H(t))$, можно записать формулу **(1)** в виде:

$$n(H,t+1) = K \cdot \frac{n(H,t) \cdot \frac{\sum_{i=1}^{n(H,t)} F(S_i^{H,t})}{n(H,t)}}{\frac{\sum_{j=1}^{N} F(S_j^t)}{N}} = n(H,t) \cdot K \cdot \frac{F_{ср}(G_H(t))}{F_{ср}(G(t))} \quad \textbf{. (2)}$$

Средняя ценность подпопуляции, соответствующей схеме **H**, может быть представлена в следующем виде: $F_{ср}(G_H(t)) = F_{ср}(G(t)) + c \cdot F_{ср}(G(t))$, где **c** - некоторая величина. Тогда формула **(2)** примет вид:

$$n(H,t+1) = n(H,t) \cdot K \cdot \frac{F_{ср}(G(t)) + c \cdot F_{ср}(G(t))}{F_{ср}(G(t))} = (1+c) \cdot K \cdot n(H,t) \quad .$$

Предположим, что величина **c** при изменении **t** не изменяется; тогда, начиная с **t=0**, получим: $n(H,t+1) = n(H,0) \cdot K \cdot (1+c)^t$, т.е. в этом случае число представителей схемы **(**строк популяции **G(t)**, соответствующих схеме**)** изменяется в геометрической прогрессии. В общем случае можно сказать, что процесс изменения представителей схемы так же аппроксимируется геометрической прогрессией.

Таким образом, в результате операции воспроизводства те схемы, для которых соответствующие подпопуляции имеют среднюю ценность выше средней в популяции, увеличивают количество своих представителей.

Воспроизводство оперирует со строками, уже присутствующими в рассматриваемой популяции, и само по себе не способно открывать новые области поиска. Для этой цели используется операция скрещивания.

Скрещивание представляет собой процесс случайного обмена значениями соответствующих элементов для произвольно сформированных пар строк. Для этого выбранные на этапе воспроизводства строки случайным образом группируются в пары. Далее каждая пара с заданной вероятностью $p_{скр}$ подвергается скрещиванию. При скрещивании происходит случайный выбор позиции разделителя **d (d=1, 2, ..., l-1**, где **l** - длина строки**)**. Затем значения первых **d** элементов первой строки записываются в соответствующие элементы второй, а значения первых **d** элементов второй строки - в соответствующие элементы первой. В результате получаем две новых строки, каждая из которых является комбинацией частей двух родительских строк.

Операция скрещивания создаёт новые строки путём некоторой комбинации значений элементов наиболее ценных в популяции **G(t)** строк. Получившиеся в результате строки могут превосходить по ценности родительские строки.

Рассмотрим некоторую схему **H**, для которой определим порядок **o(H)** - число фиксированных позиций схемы и определяющую длину **d (H)** - расстояние **(**число позиций**)** между первой и последней фиксированными позициями. Допустим, что до операции скрещивания строка **S** была представителем схемы **H**, т.е. $S \in U_H$. Допустим, что строка S^1 получена из строки **S** в результате скрещивания. Строка S^1 будет представителем схемы **H** в том случае, если позиция разделителя при скрещивании не располагалась между фиксированными позициями схемы. Вероятность того, что позиция разделителя окажется между фиксированными позициями схемы, равна: $p_d = \frac{d(H)}{l-1}$.

Учтём, что скрещивание происходит с вероятностью p_c, а также то, что даже если позиция разделителя окажется между фиксированными позициями схемы, строка S^1 может являться представителем схемы **H**, если данная строка была получена скрещиванием двух представителей схемы **H**. Тогда вероятность $p_{s,1}$ того, что строка S^1 является представителем схемы **H**, определяется выражением:

$$p_{s,1} = 1 - p_c \cdot \frac{d(H)}{l-1}.$$

Полагая независимость операций воспроизводства и скрещивания, оценим совокупный эффект от этих операций, т.е. число представителей схемы **H** в популяции **G(t+1)**:

$$n(H,t+1)/n(H,t) \cdot K \cdot \frac{F_{ср}(G_H(t))}{F_{ср}(G(t))} \cdot \left(1 - p_c \cdot \frac{d(H)}{l-1}\right).$$

Так как открытие новых областей поиска в операции скрещивания происходит лишь путём перегруппирования имеющихся в популяции комбинаций символов, то при использовании только этой операции некоторые потенциально оптимальные области могут оставаться не рассмотренными. Для предотвращения подобных ситуаций применяется операция мутации.

Мутация представляет собой процесс случайного изменения значений элементов строки. Для этого строки, получившиеся на этапе скрещивания, просматриваются поэлементно, и каждый элемент с заданной вероятностью мутации $p_{мут}$ может мутировать, т.е. изменить значение на любой случайно выбранный символ, допустимый для данной позиции. Операция мутации позволяет находить новые комбинации признаков, увеличивающих ценность строк популяции.

Допустим, что до мутации строка S^1 была представителем схемы **H**, т.е. $S^1 \in U_H$. Допустим, что строка S^2 получена из строки S^1 в результате мутации. Строка S^2 будет представителем схемы **H** в том случае, если ни один из элементов строки, соответствующий фиксированным позициям схемы, не был изменён.

Учитывая, что мутация происходит с вероятностью $p_{мут}$, вероятность p_{s2} того, что строка S^2 является представителем схемы **H**, определяется выражением: $p_{s2} = (1 - p_{мут})^{o(H)}$, где **o(H) -** число фиксированных позиций схемы **H**.

Полагая независимость операций воспроизводства, скрещивания и мутации оценим совокупный эффект от этих операций, т.е. число представителей схемы **H** в популяции **G(t+1)**:

$$n(H,t+1) \geq n(H,t) \cdot K \cdot \frac{F_{ср}(G_H(t))}{F_{ср}(G(t))} \cdot \left(1 - p_c \cdot \frac{d(H)}{l-1}\right) \cdot (1 - p_{мут})^{o(H)} \quad \textbf{(3)}$$

Так как, при малых значениях p_m приближенно можно считать $p_{s2} = (1 - p_{мут})^{o(H)} = 1 - o(H) \cdot p_{мут}$, то выражение **(3)** можно записать в виде:

$$n(H,t+1) \geq n(H,t) \cdot K \cdot \frac{F_{ср}(G_H(t))}{F_{ср}(G(t))} \cdot \left(1 - p_c \cdot \frac{\delta(H)}{l-1}\right) \cdot (1 - o(H) \cdot p_{мут})$$

или

$$n(H,t+1) \geq n(H,t) \cdot K \cdot \frac{F_{ср}(G_H(t))}{F_{ср}(G(t))} \cdot \left(1 - p_c \cdot \frac{\delta(H)}{l-1} - o(H) \cdot p_{мут}\right).$$

Таким образом, схемы, у которых малы определяющая длина и порядок, и для которых соответствующая подпопуляция имеет среднюю ценность, превышающую среднюю ценность популяции, экспоненциально увеличивают число представителей в последующих поколениях.

Очевидно, что эффективность описанной операции скрещивания значительно зависит от способа кодировки строк. Это свойство оказывается полезным для задач оптимизации функций, заданных на числовых множествах. Однако, если функция задана на произвольном множестве, например, на множестве комбинаций значений признаков объекта, где все признаки одинаковы по предпочтительности, то описанный выше способ скрещивания оказывается не вполне корректным, так как вероятность сохранения значений для групп признаков зависит от расстояния между

элементами группы в кодовой строке, а это нарушает принцип равной предпочтительности признаков. Поэтому для таких задач операцию скрещивания предполагается производить путём обмена не частями строк, а отдельными элементами. При этом задаётся некоторое число позиций n_{Π} ($n_{\Pi O}$ {1, 2, …, l}), которое определяет количество элементов строк, для которых производится обмен значениями. Число позиций n_{Π} может быть задано непосредственно или определяться случайно для каждой пары строк. Далее для каждой пары строк $(S^1, S^2)_i$, где **i -** номер пары, случайно выбираются n_{Π} номеров $n_{i,j}$ ($n_{i,j}$ O {1, 2, …, l}; jO {1, 2, …, n_{Π}}). Затем для строк пары $(S^1, S^2)_i$ производится обмен значениями элементов с номерами $n_{i,j}$, т.е. каждому элементу с номером $n_{i,j}$ строки S^1 присваивается значение элемента с номером $n_{i,j}$ строки S^2 , а элементу с номером $n_{i,j}$ строки S^2 присваивается значение элемента с номером $n_{i,j}$ строки S^1.

Допустим, что до операции скрещивания строка **S** была представителем схемы **H**, т.е. $S \in U_H$, а строка S^1 получена из строки **S** в результате поэлементного скрещивания. Вероятность $p'_{s,1}$ того, что строка S^1 будет представителем схемы **H**, равна:

$$p_{s1} = 1 - p_c \cdot \left(\frac{o(H)}{l} \right)^{n_{\pi}},$$

где **o(H) -** число фиксированных позиций схемы **H**.

Совокупный эффект от операций воспроизводства и поэлементного скрещивания, и мутации, т.е. число представителей схемы **H** в популяции **G(t+1)** определяется выражением:

$$n(H, t+1) \geq n(H, t) \cdot K \cdot \frac{F_{cp}(G_H(t))}{F_{cp}(G(t))} \cdot \left(1 - p_c \cdot \left(\frac{o(H)}{l} \right)^{n_{\pi}} \right) \cdot \left(1 - p_{мут} \right)^{o(H)}.$$

Таким образом, при поэлементном скрещивании скорость увеличения представителей схемы в последующих поколениях зависит от средней ценности схемы и количества фиксированных позиций и не зависит от расстояния между ними, а значит, не зависит от порядка расположения элементов в строке.

Итак, в результате описанных выше операций получаем **K*N N** новых строк, которые либо полностью формируют новую популяцию **G(t+1) (**при **K=1)**, заменяя при этом все строки популяции **G(t)**, либо составляют часть популяции **G(t+1)**, заменяя собой **K*N N** наименее ценных строк предыдущей популяции.

Как видно из описания алгоритма, закон $F_0(w_1, w_2, \ldots, w_n)$ вероятности распределения значений целевой функции определяется и корректируется путём использования набора **(**популяции**)** строк, содержащих наилучшие в смысле значений целевой функции комбинации элементов.